Sentiment Analysis: A New Approach for Effective Use of Linguistic Knowledge and Exploiting Similarities in a Set of Documents to be Classified.

Alekh Agarwal
Dept. of Comp Science & Engg.
I.I.T. Bombay Mumbai 400076
alekh@cse.iitb.ac.in

Pushpak Bhattacharyya
Deptt. of Comp Science & Engg.
I.I.T. Bombay Mumbai 400076
pb@cse.iitb.ac.in

Abstract

Sentiment Analysis *aims at determining the overall polarity of a document, for instance, identifying whether a movie review appreciates or criticizes a movie. We present a machine learning based approach to this problem similar to text categorization. The technique is made more effective by incorporating linguistic knowledge gathered through* Wordnet[1] *synonymy graphs. A method to improve the accuracy of classification over a set of test documents is finally given.*

1. Introduction

The field of Sentiment Analysis has been looked into at a great depth recently. There has been a lot of work in identification of polarities, subjective nature of text documents and even full-fledged ratings specially due to the potential applications([4],[2],[1]). For instance, these techniques can be employed to analyze the viewpoint of the user feedbacks. Another application is to identify and discard flames. Incorporation of such techniques in present search engines can enable users to selectively view the documents containing information just "for" or "against" a topic.

On the face of it this might sound very similar to another field being intensively researched which is of topical categorization of documents. However the two problems are quite different. The hardness of Sentiment Analysis is reflected by the failure of all the previous attempts to attain accuracies similar to those already attained in topical categorization ([2]). This mainly arises due to the fact that in Sentiment Analysis the overall sentiment may be very different from the sentiment of an individual sentence. This irony is exhibited most significantly in movie reviews. Consider a naive technique based on bag-of-words. It will most likely perform miserably on the review of a very good gory, horror movie, as such a review would be replete with words having negative sentiment in the portions where it talks about the plot of the movie. This observation is also supported by Turney's(2002)[4] work on classification of reviews.

In this paper we present a technique for the effective sentiment analysis of movie reviews. We also describe a novel approach to process the predictions for individual documents of the test dataset to improve the accuracy over the entire set. We present a *Wordnet* based method for the effective incorporation of linguistic information in our system without any kind of experts' intervention. We also present a generic method that can be used to improve the accuracy of classification over a test dataset in any kind of classification task. We show how the application of this technique to sentiment analysis helps us to attain the best accuracy so far in this field.

2. Previous Work

One of the first attempts in this field was in identifying the *genre* of texts, for instance subjective genres (Karlgen and Cutting, 1994; Finn et al., 2002). The initial approaches to sentiment detection all used linguistic heuristics, explicit list of pre-selected words and other such techniques that require use of experts' knowledge and may not yield the best possible results in all cases as pointed out in Bo Pang et al., 2002[2].

The first attempt to automate the task of sentiment classification was seen in the work of Turney(2002)[4]. He used the mutual information between a document phrase and the words "excellent" and "poor" as a metric for classification. The mutual information was determined on the basis of statistics gathered using a search engine. However, the real development in the field came with the work of Bo Pang et al.(2002)[2]. Taking the success of the supervised learning techniques in the domain of text categorization as an inspiration, they applied it to movie reviews and obtained a great improvement over the previous approaches. They

[1] http://wordnet.princeton.edu/

also introduced the concept of effective extraction of subjective sections of a document using a technique based on minimum-cuts in graphs(Bo Pang et al., 2004)[1].

3. Our Method

3.1. Setup

The core of our technique is a SVM based classifier. We made this choice because the work of Bo Pang et al.(2002)[2] clearly shows that SVMs score over all other supervised learning methods in this problem. We used bag-of-words features. We determined the strength of an adjective in a good *vs* bad classification using the *Wordnet* synonymy graph. These weights were used in place of a standard binary value in the feature vectors for SVM.

However, there is the problem of noise being introduced by the sentences that describe the plot of the movie. To deal with this we used a subjectivity detector to distinguish the parts that talk **about the movie** from those that talk about whats **in the movie**. The "about" sentences only were used for further analysis. Finally after the classification had been performed by the SVM, we took the probability estimate values from the SVM for the review being positive or negative. These values in conjunction with a similarity metric on the documents which we describe later were used to reassign the class labels in such a way that the error in classification over the whole dataset is minimized. A block diagram for the setup is shown in Figure 1.

3.2. Determining an Adjective's strength in good *vs* bad classification

We used the notion of designating strengths to adjectives in a good *vs* bad classification. The basic idea for measuring these strengths was developed using Charles Osgood's Theory of Semantic Differentiation Osgood et al.(1957,p 318)[5]. We determined the evaluative strength of adjectives using *Wordnet's* synonymy graph. A Section of this graph is given in Figure 2. Kamps et al[3] have pointed out that the evaluation function EVA(w) = $\frac{d(w,bad)-d(w,good)}{d(good,bad)}$ is an effective measure of the evaluative strength of an adjective. The geodesic function d(w_i, w_j) is given by the distance between words w_i and w_j in the *Wordnet* synonymy graph. The values are divided by d(good,bad) ,*i.e.*, the distance between the two reference words to restrict the values to [-1,1]. The JWNL[2] API was used to browse *Wordnet's* synonymy graph.

[2]http://sourceforge.net/projects/jwordnet

3.3. Detection of "about" sentences from a movie review

As pointed out earlier "about" sentences are those sentences that talk about the movie, the ones that describe the reviewer's opinion regarding the movie. These are the sentences that really help in identifying the sentiment of the movie review. We refer to these sentences as the "about" sentences and the rest of the sentences that talk of the movie plot as the "of" sentences.

Detection of *about* sentences can be approached in a similar fashion as sentiment analysis in the sense that if we train a learning algorithm on "about" *vs* "of" classification, it can be used to determine which are the "about" sentences in a document. But as pointed out in Bo Pang et al.(2004)[1], in considering just this classification, we miss out on a very valuable aspect of the information that is contained in the structural and semantic relationships among the sentences in a document.

To overcome this problem we designate two kinds of weights. The first kind is the individual weights which are the probability estimates determined by an SVM trained for "about" *vs* "of" classification. The details of computation of these estimates can be found in (Wu et al,2004)[6]. The second kind is that of mutual weights which is a measure of the tendency of two sentences to fall in the same class in a "about" *vs* "of" classification.

This tendency has two aspects. The first aspect is determined by the physical separation between two sentences in a document. Details about this are given in Bo Pang et al.(2004)[1]. The other important aspect is that of the contextual similarity between two sentences. Effective measures for including this information in Sentiment Analysis have been considered for the first time in our study. For this we used a rather crude measure based on the strength of the adjectives in the two sentences. The strengths of the adjectives are the weights that we calculated by the technique we described in Sec. 3.1. Once we have these weights, we process each sentence to compute its total adjectival strength which is just the sum of the strengths of all the adjectives in that sentence. The mutual weight is the difference between the weights of the two sentences. This value multiplied by a distance measure and appropriate scaling factors gives the final value for the mutual weight between two sentences.

Once all the individual and mutual weights have been computed, we employ a graph-cut based partition technique as described in Bo Pang et al.(2004)[1].

3.4. From individual predictions to improved accuracy on test dataset

When we described the technique for detection of "about" sentences, the point raised was that taking mutual

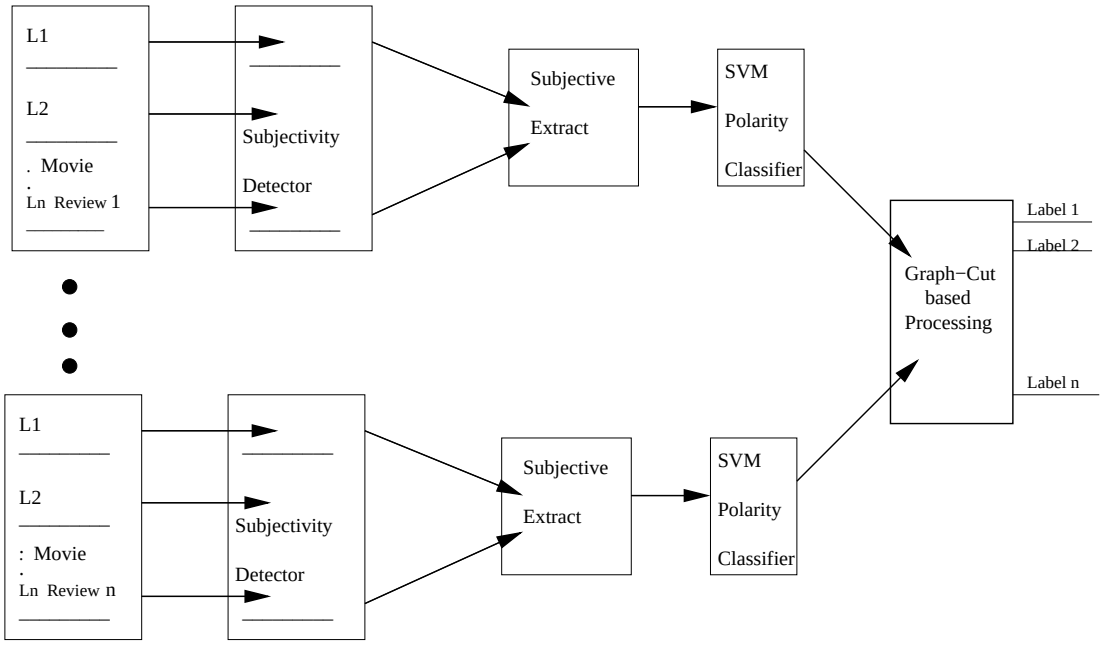

Figure 1. Flow Diagram for sentiment analysis

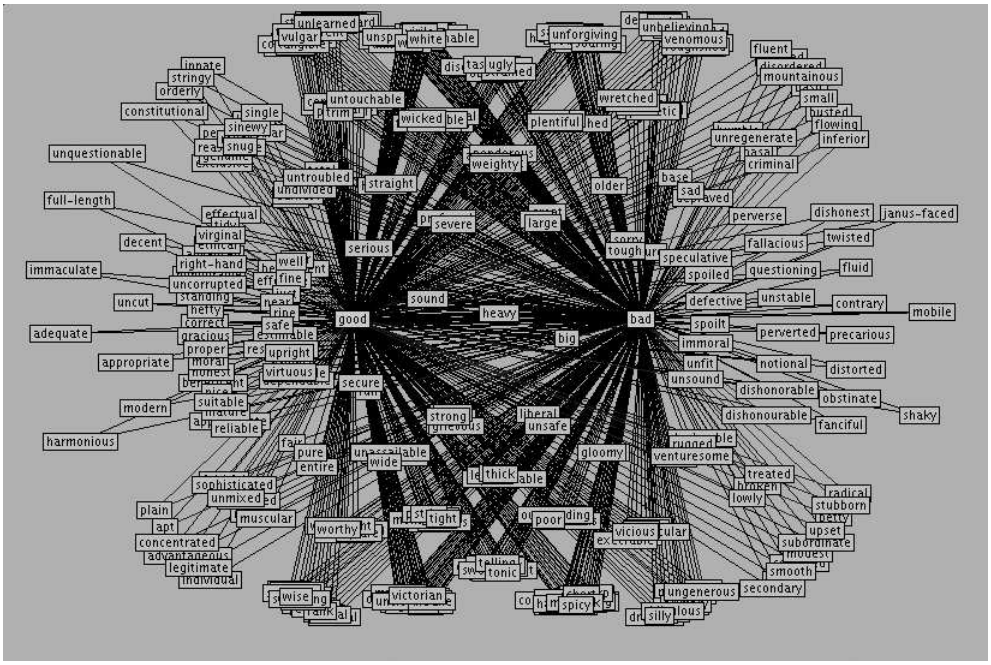

Figure 2. The Wordnet Synonym graph using "good" and "bad" as anchors upto depth 4,similarity 3

relationships into account provides a great deal of valuable information which can significantly improve the accuracy of prediction. So the very natural question that we considered was, Why don't we use these mutual relationships to increase the accuracy of sentiment analysis also if we have a set of test documents to be classified. The first task was to come up with a suitable measure for the similarity between two documents. Since feature vectors generated for the classification of document using SVMs are designed to contain a significant portion of information about the nature of the document, we calculated the number of common features in the feature vectors for every pair of documents. These values were then scaled down to [0,1]. We refer to these scaled values as the Mutual Similarity Co-efficients between a pair of documents. Hence for a pair of documents d_i and d_j, their Mutual Similarity
Co-efficient, MSC(d_i, d_j) is given by :

$$MSC(d_i, d_j) = \frac{\sum_k (F_i(f_k) * F_j(f_k)) - s_{min}}{s_{max} - s_{min}} \qquad (1)$$

where f_k is the kth feature
$F_i(f_k)$ is a function that takes the value 1 if the kth feature is present in the ith document and is 0 otherwise
s_{max} is the largest value of the number of common features between any two documents
s_{min} is the smallest value of the number of common features between any two documents.

Also the SVM was trained so as to predict the probabilities of the review being positive or negative rather than just the category label (Wu et al. [6]). The SVM probabilities when combined with Mutual Similarity Co-efficients give rise to a weights matrix on which we can apply a graph-cut partitioning technique as described in Bo Pang et al. (2004) [1]. The source and the sink nodes in this case correspond to positive and negative document respectively. The edges joining a document to source have their capacity as the probability of the document being a positive one. Similarly we assign edge capacities for the edges to sink. The edges between documents have the same capacity as the MSC of the two documents. For example, consider a set of 3 documents with values as shown in Table 1. Then the edge weights in the minimun cut setup would be as indicated in Figure 3.

	Pr_{good}	Pr_{bad}	doc1	doc2	doc3
doc1	0.231	0.769		0.314	0.867
doc2	0.642	0.358	0.314		0.421
doc3	0.301	0.699	0.867	0.421	

Table 1. An example matrix for a set of 3 documents in the minimum-cut setup

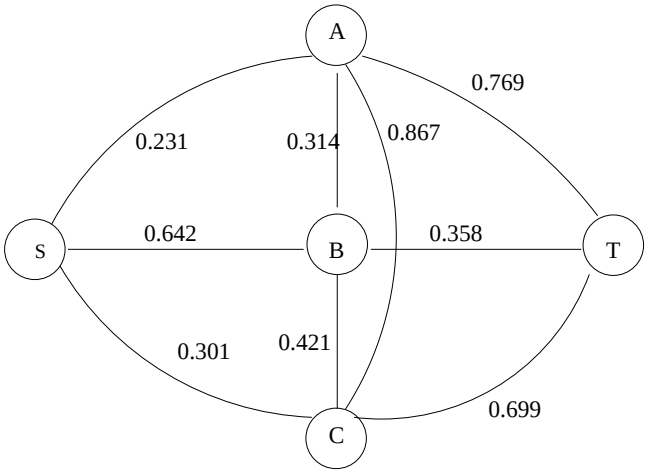

Figure 3. The graph for example described in Table 1

4. Evaluation

4.1. Experimental Setup

The movie review corpus used for this task was the tagged corpus introduced by Bo Pang et al. in ACL 2004[3]. This corpus contains 1000 positive and 1000 negative movie reviews. The entire corpus was run through a POS tagger as the POS tags were needed for later tasks. The tagger used was Stanford Log-Linear Model Tagger v1.0[4]. The resulting documents were used for subjectivity detection.

Detection of *about* sentences This module had two parts. The first was for the estimation of individual weights. Here we used an SVM that was trained to predict probability estimates rather than class labels. The dataset used was the Subjectivity Dataset introduced by Bo Pang et al. in ACL 2004. This corpus contains 5000 movie-review snippets and 5000 plot summaries. The SVM package used was libsvm-2.71[7].

The second module was to calculate the mutual weights. Here we used the Ford-Fullkerson algorithm to obtain the minimum-cut. For calculation of the mutual weights we experimented with a number of measures. Consider two sentences d lines apart. Let w_x and w_y be the weights of the sentences as obtained by summing the strengths of all the adjectives in the sentences x and y *resp*. For example, if x is the sentence "The movie was excellent with outstanding performances from all actors", then w_x would be

[3]Corpus available at www.cs.cornell.edu/people/pabo/movie-review-data (review corpus version 2.0).

[4]http://www-nlp.stanford.edu/software/tagger.shtml

$w_{excellent}$+$w_{outstanding}$.

Let $assoc(s_i, s_j)$ be the mutual weight for the sentence pair s_i and s_j. Then we have

$$assoc(s_i, sj) = c * f(d) * g(w_i, w_j) \quad (2)$$

We experimented with $f(d) = 1$ and $f(d) = \frac{1}{d^2}$.
Similarly, we tried out $g(w_i, w_j) = |w_i - w_j|$ and $g(w_i, w_j) = 1$.

c is just a constant factor. A larger value of c implies that the algorithm will be more loath at putting sentences not having a great deal of similarity in different classes. Different values of c were tried out with an aim of optimizing the classification results downstream.

5. Experimental Results

As the first step we proceeded to run our tests on the whole documents without taking any extracts of "about" sentences. We tried out different kinds of features. Our first guess was that since adjectives play a very important role in determining the polarity of a document, using only adjectives as the features should be sufficient. We used the BNS feature selection algorithm (Forman 2003[9]) to select the top 16000 adjectives. In the first approach we took the adjective weights described in Sec. 3.1 and multiplied them with appropriate multipliers if any modifiers were present before them. If the adjective was lying between a "not" and a punctuation mark, then the weight was multiplied by -1. We took the summation of these weights for all the occurrences of each feature in a document. Please note that "good", "very good" and "not good" are all considered to be instances of the same feature "good". The weights for these would be w_{good}, $m_{very} * w_{good}$ and $-1 * w_{good}$ *resp.* where m_x denotes the weight of the modifier x. Using these feature vectors we obtained a five-fold cross validation accuracy of 68.1% over the dataset.

We then took the top 32000 unigrams as our features. The adjectives in these were filtered out for a separate treatment. For other features we used just binary values, *i.e.* 1 if the feature is present and 0 if the feature is absent. For adjectives, we tried out the same approach as earlier. The five-fold cross validation accuracy in this case was found to be 70.2%.

An observation made at this stage was that some features like "better" appear both as an adjective and not as an adjective. We decided to prefix "ADJ_" to all the adjectives to have an elementary Word-sense Disambiguation. We also prefixed a "NOT_" to every occurrence of a feature that occurs between "not" and a punctuation mark. Using summations of weights for adjectives and binary values for other features, the accuracy was found to be 68.29%.

We then did a finer classification for adjectives. Each occurrence of an adjective after a negative modifier was labelled with "NEG_ADJ_" tag and that following a positive modifier was labelled with "POS_ADJ_" tag. For example, the phrases "not very good" and "very good" would correspond to the features "NEG_ADJ_good" and "POS_ADJ_good" *resp.* The absence of any modifier was also treated as a positive modifier. We took the largest absolute value of the weight over all the occurrences of each feature. (*i.e.* weight of strongest negative modifier present*adjective weight for NEG_ADJ_ and weight of strongest positive modifier present*adjective weight for POS_ADJ_). With these features the accuracy dropped down to 65.5%.

We finally took the top 32000 unigrams and separated the adjectives as earlier. This time we just used the weight of the adjective in the feature vector of the document if the adjective was present. For other features still the binary values were used. With these features, the accuracy of classification was found out to be 75.8%.

As the last experiment was found to give the best results, we decided to proceed with this approach for further experiments with documents filtered from after detection of the "about" sentences. We first chose $f(d) = \frac{1}{d^2}$ and $g(w_i, w_j) = |w_i - w_j|$. Two different values of c, 10 and 100 were tried out. For $c = 10$, the classification accuracy was 70%. For $c = 100$, an accuracy of 65.65% was obtained.

This is in agreement with the intuition that a lower value for c should yield better results. To further confirm this hypothesis we tried $c = 1$. For this case the average five-fold cross validation accuracy was 67.5%.

This perhaps shows that too strict a penalty for a dissimilarity or distance between sentences also leads to a decline in accuracy. But a decline of accuracy on use of "about" extracts was counter to the expectations. This is probably due to the crude function that was used to model sentence similarities. A better measure can be expected to give better results. We chose such a simple function because the complexity involved in the computation of statistically reliable functions like the Mutual Information Quotient appeared to be prohibitive in this case.

We then experimented by using distance and contextual similarity in isolation. With just a distance measure and $c = 100$, we obtained an accuracy of 65.8%. In the same case, using $c = 10$ gave an accuracy of 68%.

Using just the contextual similarity measure gave an accuracy of 68% both for $c = 10$ and $c = 100$.

Till this point we hadn't taken into account the fact that using the mutual similarities between the documents can be used to find out the problems with current predicted labels and can thus provide a significant increase in accuracy. We decided to apply this technique described in Sec. 3.3 to the results obtained from all the previous steps.

For complete documents using weights for adjectives

	Type of documents	Before Graph-cut	After Graph-cut
1.	Full documents	75.8%	95.6%
2.	"about" extracts with distance and context info $c = 100$	65.65%	94.2%
3.	"about" extracts with distance and context info $c = 10$	70%	92%
4.	"about" extracts with distance and context info $c = 1$	67%	93.5%
4.	"about" extracts with distance info $c = 100$	65.8%	91%
5.	"about" extracts with distance info $c = 10$	68%	89.4%
6.	"about" extracts with context info $c = 100$	68%	84.2%
7.	"about" extracts with context info $c = 10$	68%	84%

Table 2. Five-fold cross validation accuracies for various experiments

and binary values for other features, application of this technique improved the accuracy to an overwhelming 95.6%. All the results before and after the application of this technique are listed in Table 2. We also tried out the use of BNS feature selection algorithm but no significant change in results was observed.

6. Conclusions

Clearly, the main strength of our approach lies in showing how strong an influence mutual relationships between documents can have on their sentiment analysis. The way in which we have used the graph-cut technique for this task provides a very simple yet efficient framework for incorporating this information. Moreover, this technique can be applied to improve the accuracy of predictions in any classification task over a set of test documents.

However, one observation counter to intuition as well as to prior study (Bo Pang et al. 2004[1]) has been the decline in accuracy of classification upon using the subjectivity detection technique.

But the accuracy obtained in our approach still scores over those obtained in the previous approaches. This is the first time sentiment analysis has been possible with an accuracy of over 90%. The compactness of the subjective extracts provides a great reduction in the processing time. With a very little difference in the final accuracies, it can be considered as a valuable trade-off in the practical applications. Also, as these jobs are usually done in batch mode, it will not be very uncommon that a real application has to classify a set of documents rather than a single document. And that is where this technique does really well.

The main contribution of our work is in two main directions. Firstly, *Wordnet* as a lexical resource is taking more and more prominence. We have demonstrated how the structure of links from *Wordnet* can be applied to tasks where human experts' intervention has been considered inevitable in the past. So this is a major step in the automated mining of valuable linguistic information.

Secondly, the manner in which we use graph-cut technique to improve the accuracy of classification provides a very generic method for obtaining better results in any problem that can be modelled in the framework. The technique is inherently simple yet extremely powerful as demonstrated by our results.

Future work consists in looking for better measures to incorporate deeper linguistic information in the approach. The *Wordnet* based techniques that we used provide a very convenient framework to automate the mining of such information. Better measures for representing the degree of similarity of sentences and a separate treatment for other Parts-of-Speech like we did for adjectives can also be considered. Adverbs might be very interesting to look at as they turned out to be the most frequent adjectival modifiers from what we realized during our work.

7. Acknowledgments

We would like to thank Nitin Agarwal, Parijat Garg, C.-J.Lin, Bo Pang, Prof. S. ArunKumar, Harsh Jain, Vijay Krishnan and all other friends and colleagues who gave us helpful suggestions during the course of work.

References

[1] Bo Pang and Lillian Lee, *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*, Proceedings of ACL, 2004.

[2] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan, *Thumbs Up? Sentiment Classification Using Machine Learning Techniques*, Proceedings of EMNLP 2002,pp 79-86.

[3] Jaap Kamps, Robert J. Mokken, Maarten Marx, and Maarten de Rijke. *Using WordNet to measure semantic orientation of adjectives.* In Proceedings of

the 4th International Conference on Language Resources and Evaluation (LREC 2004), volume IV, pages 1115-1118. European Language Resources Association, Paris, 2004.

[4] Peter Turney. 2002. Thumbs up or thumbs down? *Semantic orientation applied to unsupervised classication of reviews.* In Proc. of the ACL.

[5] Osgood, C. E., G. J. Succi, and P. H. Tannenbaum, 1957. *The Measurement of Meaning.* University of Illinois Press, Urbana IL.

[6] T.-F. Wu, C.-J. Lin, and R. C. Weng. *Probability estimates for multi-class classification by pairwise coupling.* In S. Thrun, L. Saul, and B. Scholkopf, editors, Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA, 2004. URL http://www.csie.ntu.edu.tw/ cjlin/papers/svmprob.pdf.h

[7] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001.Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[8] Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1990. Introduction to Algorithms. MIT Press.

[9] George Forman *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*, Journal of Machine Learning Research 2003, pages 1289-1305.